Problem A. Arithmetic

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Manao loves integer numbers very much. For the last few months he spent all his time inventing and solving different, sometimes senseless, equations. He is not that good in mathematics to solve all the equations he invented, though. He therefore has a long list of unsolved equations. Fortunately, he heard that students who participate in ACM competitions are perfect in mathematics, and decided to ask help from you.

Your task is to help Manao find a positive integer x such that $a^x + b^x$ is divisible by p^k. Here, a, b, k and x are positive ingegers.

Input

The input contains a single line with four integers a, b, p and k. It is guaranteed that p is and odd prime, $1 \le a, b, p \le 10^4$, $0 < k < 38$ and $1 \le p^k \le 10^{18}$.

Output

Your program should output a single line with an integer $1 \le x \le 10^{18}$, the solution to Manao's equation. If there is no such x, you should output -1 instead. If there are more than one solution, print arbitrary one.

Examples

standard input	standard output
3 4 5 2	2
1 1 3 1	-1

Explanations

In the first example, $3^2 + 4^2 = 25$ is divisible by $5^2 = 25$.

In the second example, $1^x + 1^x$ is always equal to 2, so there is no solution.

Problem B. Cantonal Courier

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

To earn money for the journey of your dreams (Valencia and Yekaterinburg) you come up with an idea for a shipping company. You are going to have an employee (a courier) in each canton of Algoland. The caveat that is going to let you offer competitive rates while promoting sustainability is that your couriers are going to use public transport. The distinctive feature of Algoland's public transport system is that each canton consists of several zones — depending on your route you need to buy the tickets for some subset of those zones.

For each canton you are given a list of possible assignments and the reward for each of them that a customer will pay if you agree to take it. For each assignment you are also given the list of zones for which you need a ticket in case you want to do it. Finally, you are given the price of the ticket for each zone. All the tickets are day passes and your couriers are truly excellent, so you can be sure that once bought, a single ticket can be reused for several jobs.

Find the optimal profit (payment from jobs minus costs of tickets) you can achieve for each canton.

Input

The input describes a single canton and starts with a line holding two integers Z and J where Z is the number of zones and J is the number of jobs ($1 \le Z, J \le 100$). In the next line there are Z integers $c_1, \ldots, c_Z$ where c_i is the cost of the ticket for zone i ($1 \le c_i \le 5000$). The third line contains J integers $p_1, \ldots, p_J$ where p_i is the reward for job i ($1 \le p_i \le 5000$).

J lines follow — i-th of those lines describes the tickets needed for job i. Each of those lines starts with $0 \le N_i \le Z$, followed by a strictly increasing sequence of N_i zones (1-based) for which the tickets are needed.

All consecutive numbers in a single line are single-space separated.

Output

Print a single line with an integer: your profit given optimum choice of jobs.

Examples

standard input	standard output
4 3 1 5 6 7 3 4 10 2 1 2 1 2 2 3 4	1
3 3 3 3 3 4 4 4 3 1 2 3 3 1 2 3 3 1 2 3	3

Problem C. Merlin's Orb

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Old Merlin owns a powerful crystal ball, which allows him to chat with his fellow wizards wirelessly. Unfortunately for our friend, not so long ago the crystal orb started to exhibit a very strange behaviour. Instead of showing a smooth crystal-clear image as usual, the ball now displays some annoying greenish circles on its surface. To make matters worse, at any intersection point of two circles a potentially destructive high-energy beam of light spews out of the ball. Understandably, Merlin is terribly worried about these leaks, as too many of them would cause his prized orb to disintegrate in an Earth-shattering explosion (an event highly undesirable).

There is a glimmer of hope, however: after carefully studying the problem, he is now able to pre-determine the size of the circles, but not their positions, so the circles will now have a known size each, but they will appear uniformly and independently at random on the surface of the crystal sphere.

Merlin now wants to devote all his time repairing his device and needs some help from you. He would like to have an idea about how much more damage the orb will sustain, so he wants you to evaluate the expected number of energy releases that will occur during the time he is working on the orb. It's also good to know that in the extremely unlikely event that two circles overlap completely, nothing happens (Merlin tried this).

Merlin is a nice wizard and wants to save you some time so he provides you with one with his many theorems: Suppose you have a sphere, and that the equator of this sphere has latitude 0, and latitude increases towards the north pole, which has latitude 90 degrees. Then the area of surface that lies north of the circle of latitude φ is $2\pi R^2(1-\sin\varphi)$. He also gives you an old book of trigonometry: after you open it on a random page, you see "$\cos(\alpha-\beta)-\cos(\alpha+\beta) = 2\sin\alpha\sin\beta$".

Input

There are two lines of input.

The first line contains two integers: the number N of circles that will appear, and the radius of crystal ball, given as an integer R, where $2 \le N \le 3 \cdot 10^5$ and $0 < R \le 10^{18}$.

The second line contains N integers $0 < r_i \le R$, $1 \le i \le N$, representing the radii of the circles.

Output

Print one line containing just one real number, representing the expected number of points of circle intersections.

Your answer will be accepted if either the relative error or the absolute error is less than 10^{-7}.

Examples

standard input	standard output
2 5 5 5	2.0
3 7 1 2 4	0.571428571

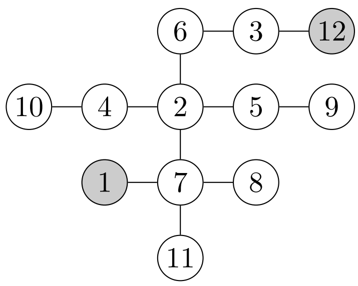
Problem D. Multidrink

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

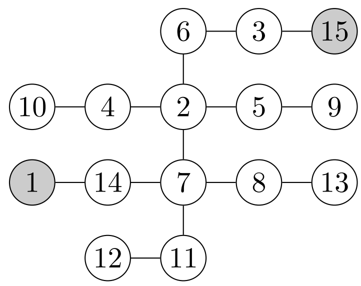
Byteasar lives in Byteburg, a city famous for its milk bars on every corner. One day Byteasar came up with an idea of a "milk multidrink": he wants to visit each milk bar for a drink exactly once. Ideally, Byteasar would like to come up with a route such that the next bar is always no further than two blocks (precisely: intersections) away from the previous one.

The intersections in Byteburg are numbered from 1 to n, and all the streets are bidirectional. Between each pair of intersections there is a unique direct route, that is, one that does not visit any intersection twice. Byteasar begins at the intersection no. 1 and finishes at the intersection no. n.

Your task is to find any route that satisfies Byteasar's requirements if such a route exists.



An exemplary route satisfying the requirements is: 1, 11, 8, 7, 5, 9, 2, 10, 4, 6, 3, 12.



There is no route that satisfies the requirements.

Input

In the first line of input there is a single integer n ($2 \le n \le 500\,000$), denoting the number of intersections in Byteburg. Each of the following $n-1$ lines holds a pair of distinct integers a_i and b_i ($1 \le a_i, b_i \le n$), separated by a single space, that represent the street linking the intersections no. a_i and b_i.

Output

If there is no route satisfying Byteasar's requirements, your program should print a single word "`BRAK`" (Polish for *none*), without the quotation marks to the standard output. Otherwise, your program should print n lines to the standard output, the i-th of which should contain the number of the i-th intersection on an arbitrary route satisfying Byteasar's requirements. Obviously, in that case the first line should hold the number 1, and the n-th line — number n.

Examples

standard input	standard output
12 1 7 7 8 7 11 7 2 2 4 4 10 2 5 5 9 2 6 3 6 3 12	1 11 8 7 4 10 2 9 5 6 3 12
10 1 9 9 6 1 2 2 8 10 7 7 3 10 5 5 4 1 10	BRAK

Problem E. New Year

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

New Year is just around the corner, and wizards at the Unseen University of Zurich want their hallways to be as colorful as possible for this occasion. They have spells for K different colors (numbered from 1 to K) and there is a total of N chambers in the University, $\lceil N/2 \rceil$ in the west and $\lfloor N/2 \rfloor$ in the east wing.

Each hallway connects a chamber in the west wing with a chamber in the east wing and no two hallways connect the same pair of chambers. Since colors are magical (as everything else in that place), they tend to become unstable and possibly explode if two adjacent hallways have the same color. As you are an expert in colors and hallways at the Adventure consulting company, wizards have hired you to find out the largest number of hallways they can color such that no two colors are too close to each other, that is, no two hallways which leave the same chamber have the same color. Since this number alone is not very helpful for them, they also asked you to provide them with one such coloring.

All hallways that are not colored will be completely dark and depressing, and wizards want their University to be as little depressing as possible for the New Year! If you fail to find the answer, or you provide them with a wrong one, they will turn you into a frog!

Input

Input starts with a line holding three integers: N ($1 \le N \le 200$) — the number of chambers, M ($2 \le M \le 10^4$) — the number of hallways, and K ($1 \le K \le 20$) — the number of different colors. Chambers are numbered from 1 to N. In each of the next M lines there are two integers a and b ($1 \le a \le \lfloor N/2 \rfloor$, $\lfloor N/2 \rfloor + 1 \le b \le N$), specifying there is a hallway running from chamber a to chamber b.

Output

First print a number L, the largest number of hallways that can be colored such that the condition from the statement holds, and in the next L lines output numbers h and c — meaning that the h-th hallway from the input should be colored with color c. Any valid coloring will be accepted!

Example

standard input	standard output
8 8 2 1 5 2 6 3 7 4 8 1 6 2 7 3 8 4 7	7 1 1 2 1 3 1 4 1 5 2 6 2 7 2

Problem F. Packing T-Shirts

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are currently packing your luggage for your trip to Valencia, and like any self-respecting programmer you must face the difficult task of choosing which T-shirts to take with you. After having packed all other bare essentials (laptop, mouse, keyboard, socks, etc.) and having reserved some space for all the paraphernalia you will carry back from SWERC (medal, diploma, your SWERC T-shirt, prizes), you are left with a capacity of c in your luggage.

While the weights of your T-shirts look random to your teammates, they in fact obey the arcane rule: you have chosen a superincreasing sequence $a = (a_1, a_2, \ldots, a_n)$ (superincreasing means that each element is larger than the sum of all the previous elements: $a_i > \sum_{j=1}^{i-1} a_j$); a prime number q larger than the sum of all a_i; and an integer r that is coprime to q. The weight of your i-th T-shirt in your collection is equal to $r \cdot a_i \pmod q$.

You would like to choose a set of T-shirts to take with you, such that their combined weight is exactly equal to c. This will impress your teammates, who will think that you can solve NP-complete problems.

Input

Input consists of no more than 20 test cases.

Each test case starts with a line holding three integers: N ($1 \le N \le 61$), q ($2 \le q \le 2^{62} - 57$, q is prime) and r ($2 \le r < q$). On the next line there are N integers $w_1, \ldots, w_n$, the weights of your T-shirts. On the next line there is an integer c ($1 \le c \le 2^{62} - 1$), the remaining capacity of your luggage.

It is guaranteed that $w_i = r \cdot a_i \pmod q$ for some a_i, the sequence $(a_1, \ldots, a_n)$ is superincreasing, and $\sum_{i=1}^{N} a_i < q$). We also guarantee you that there is always at least one solution for each given test case.

The input terminates with a line containing three zeros. This line must not be processed.

Output

For each test case, output a space-separated list of integers S_1, S_2, ... on a single line, such that $(S_1, S_2, \ldots)$ is a subset of $(w_1, \ldots, w_n)$, the sum of elements in that list equals the capacity ($\sum_{i=1}^{|S|} S_i = c$), and such that the list is sorted ($S_i < S_{i+1}$).

Example

standard input	standard output
8 881 588 295 592 301 14 28 353 120 236 1129 0 0 0	236 301 592

Problem G. Fancy Runway

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are approaching Valencia airport, as you glance out of the window and spot the curious illumination of the runway: a line of N red or yellow lamps indicates that you are probably in the right place. Each second, the colours of the lamps change. Lamp i takes the colour that lamp $i+1$ had just before, giving the impression of a sliding ground! Although it looks random, the colour of the N-th lamp is determined in a deterministic way and depends on a subset S of lamps, which always includes lamp 1. If an odd number of those subset lamps currently shine red, then the N-th lamp will be red in the next state and yellow otherwise. Your coach, who knows lots of mazy stories, tells you that Spanish pilots read this colour code as instructions for landing approaches! Given the current state and the state at which your plane shall touch down, can you compute how long you will still be hovering over Valencia?

Input

Input consists of no more than 10 test cases.

Each test case consists of three lines: The first line gives N ($2 \leq N \leq 32$), the number of lamps, followed by S, the number of lamps in the subset ($2 \leq S \leq N$) steering the N-th lamp.

Line two holds S integers, indicating which lamps are part of that subset. Line three provides two N-character-long strings with characters '`r`' (for red) and '`y`' (for yellow), describing the current and final states. The first character of the string describes lamp 1, the last one lamp N.

The input terminates with a line containing two zeros. This line must not be processed.

Output

For each test case, output a single integer: the minimum amount of time (in seconds) it will take your plane to touch down if this is possible given the current and final states. Otherwise, print a line containing the single character '`*`'.

Example

standard input	standard output
2 2 1 2 rr ry 4 2 1 2 ryry rryy 7 2 1 3 yryrryy yrrrryy 0 0	1 6 *

Problem H. Scouting Camp

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Jonas and his friends are organizing a scouting camp. A friendly farmer gave them permission to construct their tents on his meadow. This they did. To protect the meadow during the camp, they now want to lay out a road network that connects all the tents using Passareco elements.

These Passareco elements are wooden paths that can be laid down in either east-west or north-south direction on the camping ground. For example, to connect the tent at $(4,7)$ with the tent at $(19,3)$, one would need 15 meters of Passareco going east followed by 4 meters going north (or the other way around), for a total of 19 meters. The problem is, it sometimes rains in scouting camps. Thus, Jonas' friends would like to construct the Passareco network in a way that minimizes the length of the longest Passareco segment. This means they can move from any tent in the camping ground to another, taking shelter at tents on their way, without staying in the rain for too long.

Input

The input consists of $N+1$ lines.

The first line contains N, the number of tents ($1 \leq N \leq 2 \cdot 10^4$). The following N lines each contain two integers x and y which denote the position of a tent. You may assume that $|x|, |y| \leq 2 \cdot 10^7$. No two tents are placed in the same point.

Output

Print a single number on a line by itself: the largest distance between two tents connected by an uninterrupted Passareco way.

Example

standard input	standard output
5 1 3 4 1 5 3 8 4 4 6	4

Explanation

The sample input corresponds to the following layout:

```
      2
      |
      .
      |
1-.-.-.-3
        |
        .-.-.-4
        |
        .
        |
      5-.
```

The longest way out in the rain is 4 meters long. For example, when going from tent 1 to tent 5, one can make a detour to tent 3 in order to minimize the longest consecutive time period spent under the rain.

Problem I. Sequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

You are given a sequence of N integers. Your task is to split it into blocks of consecutive elements in such a way that the blocks' sums form an increasing sequence of maximum length.

For example the sequence 6, −5, 11, −5, 8 can be split as $(6, -5)$, $(11, -5)$, (8) issuing the sums $1, 6, 8$. It is clear that increasing sequence of length 4 or more can't be constructed.

Input

The first line of input contains the number N ($1 \leq N \leq 4000$). Each of the next N lines contains an integer a_i, the i-th element of the initial sequence. Each element doesn't exceed 10^8 by absolute value.

Output

In the first line output an integer B, the number of blocks the initial sequence should be divided into. In the second line output exactly $B - 1$ integers which describe an optimal partition. These integers should be 0-based indices of the first elements of blocks 2, 3, …, B. It means that the first number must be the 0-based index of the first element of the second block, the second number must be the index of the first element of the third block, etc., the last number must be the index of the first element of the B-th block.

If $B = 1$, the second line must be empty.

If there are multiple solutions, output any of them.

Examples

standard input	standard output
5 6 -5 11 -5 8	3 2 4
3 4 5 6	3 1 2
3 5 4 3	2 1
3 3 2 1	1

Problem J. Sliding Puzzle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You may be familiar with the so called sliding puzzle. If not, don't worry, here is a short description:

- There are sliding puzzles of different sizes. The puzzle of size N is played on an $N \times N$ square and it will hold that N is positive integer, not less than 2.
- The fields are filled with the numbers 1 through $N^2 - 1$ in some given starting configuration. Each number appears exactly once, hence there is exactly one free field.
- A valid move consists of moving one of the neighbors of the free field to the free field itself. That is, in each configuration you have between 2 and 4 valid moves and a move can be specified by the number which will end up in the free field. (A valid move can alternatively be seen as moving the free field to one of its neighbors.)

In this task you will be given a starting configuration for a sliding puzzle of size N, as well as M moves. You have to simulate all the moves and output the resulting configuration. Invalid moves should be ignored.

Input

The input consists of at most 10 test cases.

Each test case starts with a line of two integers N ($2 \le N \le 100$), the size of the sliding puzzle, and M ($0 \le M \le 10^5$), the number of moves to simulate.

The next line contains a permutation of the integers 0 through $N^2 - 1$, separated by spaces. It describes the starting configuration. The number on field (x, y), $1 \le x, y \le N$, is given as the $((y-1) \cdot N + x)$-th element of the permutation. The number 0 represents the free field. The top-left corner has coordinates $(1, 1)$.

Each of the following M lines contains an integer X ($1 \le X \le N^2 - 1$), describing a move as explained above. Remember that invalid moves should be ignored.

The input terminates with a line containing two zeros. This line must not be processed.

Output

For each test case, output a single line with the sliding puzzle configuration after simulating the M moves, in the same format we used in the input. That is, you should output a single line containing a permutation of the numbers 0 through $N^2 - 1$.

Example

standard input	standard output
3 6 7 4 0 3 1 6 8 5 2 4 3 1 3 7 1 0 0	1 0 4 7 3 6 8 5 2

Problem K. Transport

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 seconds
Memory limit:	256 mebibytes

There are N cities in Byteland. They are connected by the railway roads in the following way:

- There are R pairs of cities connected by direct railroad without any intermediate stations. The fare is equal to a bytelars for any such passage. It is allowed to pass along these roads in both directions.
- It is possible to get from any city of Byteland to another one using the railway system.

«Byteland Airways» decided to create several bidirectional flights. These flights are established only between such pairs of cities, that the cheapest way to get from one of them to another by the railway has *exactly* one interchange. The fare of any flight is equal to b bytelars.

Byteasar is a candidate for mayor of city M. He intends to create a table containing minimal travel costs between city M and all other cities in Byteland. You have to do it for him.

Input

The first line of input contains five integers: N, R, M, a and b ($2 \le N \le 10^5$, $1 \le R \le 10^5$, $1 \le M \le N$, $1 \le a, b \le 1000$). They mean the number of cities in Byteland, number of railroads, the city for which it is required to calculate the table of minimal travel costs, the fare of unit railway passage, and the fare of flight respectively.

Each of the following R lines contains two integers: u_i and v_i ($1 \le u_i, v_i \le N$, $u_i \ne v_i$ for $i = 1, 2, \ldots, R$). They mean the numbers of cities connected by the i-th railroad. Any two cities are directly connected by at most one railroad.

Output

Output N lines: i-th of them should contain the minimal cost of travel from city M to city i. In particular the M-th line should contain the number 0.

Example

standard input	standard output
5 5 1 6 4 2 1 3 2 4 3 1 3 5 1	0 6 6 4 6

Problem L. Walk Through Byteland

Input file:	*standard input*
Output file:	*standard output*
Time limit:	30 seconds
Memory limit:	256 mebibytes

Each city in Byteland has the name which looks like a binary string of length n. Each bit in the name can be only 0 or 1. Obviously different cities have different names. There are $2^n - k$ cities in Byteland. It means that there are exactly k binary words of length n, which are not used as the names of cities.

Some pairs of cities are connected by roads. There is a road between two cities if and only if their names differ only in one bit.

Your task is to determine whether there is a route from city i to city j by the roads of Byteland.

Input

The first line of input contains two integers n and k ($1 \le n \le 60$, $0 \le k \le 1\,000\,000$, $k < 2^n - 1$, $n \cdot k \le 5\,000\,000$) which mean the number of bits in cities' names and the number of binary strings of length n not being used as the names of cities. The second line contains two binary strings of length n which are the names of cities i and j. Each of the next k lines contains a binary string of length n, which is not used as the name of any city in Byteland. All these k strings are different. It is guaranteed that the names of cities i and j do not appear among these k words.

Output

If there is a path from city i to city j, output only the word "TAK". Output the word "NIE" otherwise.

Examples

standard input	standard output
4 6 0000 1011 0110 0111 0011 1101 1010 1001	TAK
2 2 00 11 01 10	NIE